

SW Engineering CSC 648 Summer 2021

DormMates

Milestone 3 • Version 2 • 30 July 2021

Team 01

Team Lead and Github master	Andrei Georgescu	ageorgescu@mail.sfsu.edu
Frontend Lead	Meeka Cayabyab	
Backend & Database Lead	Jonathan McGrath	
Engineer	Alexandre Ruffo	
Engineer	Jimmy Yeung	
Engineer	Sayed Hamid	
Engineer	Ahad Zafar	

History Table

Version	Date	Notes
M3V2	08/01/2021	Addressed wireframe and diagrams feedback.
M3V1	07/22/2021	Initial submission
M2V2	07/19/2021	Addressed the vertical prototype, functional requirements, and diagrams feedback.
M2V1	07/08/2021	Initial submission
M1V2	07/01/2021	Added database master role to the title page, addressed use cases feedback, addressed functional requirements feedback, and addressed competitive analysis feedback.
M1V1	06/22/2021	Initial submission

Table of Contents

Table of Contents	3
Data Definitions V3	4
Functional Requirements V3	7
Priority 1 - Required	7
Priority 2 - Desired	8
Priority 3 - Opportunistic	9
Wireframes	10
Use Case 1: Creating a Landlord Account and Posting a Listing	10
Use Case 4: Student Creates a New Account	11
Use Case 5: Student Needs Roommates	12
All Wireframes	13
High Level Database Architecture and Organization V2	14
Database Model	14
High Level Diagrams V2	15
UML Diagram	15
Network Diagram	16
Deployment Diagram	17
Detailed List of Contributions	18

Data Definitions V3

- **General User:** A user who can view and access the website limited to the website's features. A general user must **register** by creating a student or landlord account to gain more privileges.
 - **Registration:** A general user has the option to create an account.
 - A general user shall be able to view limited listings based on an institution.
 - A general user shall be able to view the informational pages regarding the service.
- **Registered User:** A user who has access to the website's core features.
 - **Account Contains:** All registered user accounts must contain the following information.
 - **Username:** A username is required to create an account.
 - **Email:** A unique email is required to create an account.
 - **Password:** A password that will be hashed for security purposes is required.
 - **Accepted Terms of Use:** Must accept to create an account.
 - **Name:** A registered user must have a first and last name.
 - **Gender:** A registered user must specify a gender.
 - **Age:** A registered user must have an age.
 - **Photo:** A profile shall have a picture of the registered user.
 - Has the ability to log into the service.
 - **Email:** Needs to login with a unique email.
 - **Username:** Needs to login with a unique username.
 - **Password:** Needs to provide a password.
 - Has the ability to communicate with other users.
 - Has the ability to create a student account.
 - Has the ability to create a landlord account
 - **Student:** A user who signed up with a .edu email is denoted a student.
 - Has the ability to create a group.
 - Has the ability to invite other student users to a group.
 - Has the ability to invite landlords to groups.
 - **Profile:** A profile will display information / attributes of a student user.
 - **Institution :** An institution will be part of a profile
 - **Major:** A major will be part of a student profile
 - **Schedule:** Needs to specify if they have a night or day schedule
 - **Hobby:** A hobby will be part of a student profile.
 - **Personalities:** A personality will be part of a student profile.
 - **Social Media:** A link to social media outlets.

- **Landlord:** A user who signed up with a non .edu email is denoted as a landlord.
 - Has the ability to post new listings.
 - Has the ability to create leases.
 - Has the ability to add students to a lease.
 - **Profile:** A profile will display information / attributes of a landlord.
 - **Listings:** The listings that they have posted
 - **Rating:** A list of ratings given by students.
 - **Student Ratings:** Number of students that have rated the landlord.
- **Admin User:** User that has the ability to moderate the service.
 - Can validate listings and make them public on the platform.
 - Can remove listings from the platform.
 - Can answer reports made by other users on the platform.
 - **Account Contains:** All registered user accounts must contain the following information.
 - **Username:** A username is required for all admin users.
 - **Email:** A unique email is required for all admin users.
 - **Password:** A password that will be hashed is required for all admin users.
- **Listing:** A representation of a housing unit that was posted by a landlord.
 - **Location:** Represents the physical location of a listing.
 - Represented by latitude and longitude coordinates.
 - **Photos:** Represents visual representations of a listing.
 - **Lease:** Represents an agreement between one or more students and a landlord which signifies that a lease between the parties exists.
 - Landlords can create leases for their listings.
 - Students can sign leases.
 - **Amenities:** Represents the key features of a listing that are included.
 - Washer and Dryer
 - Wifi
 - Closets
 - Bathrooms
 - Furnished
 - Whiteboard
 - Living room
 - Kitchen
 - Patio
 - Parking
- **Favorite:** Specific listings that a user is interested in and wants to bookmark.

- **Match:** A representation of two student users who may be compatible roommates.
- **Groups:** A representation of a group of students that are matched and looking for listings together.
 - **Chat:** A way groups can communicate with each other on our platform.
 - Student users can share listing links in chat.
 - Student users can add a landlord user to a chat.
 - Landlord users can create a lease in a chat and invite all student users to sign the lease.

Functional Requirements V3

1. Priority 1 - Required

1.1. Unregistered User

- 1.1.1. An unregistered user can create a new student or landlord account.
- 1.1.2. An unregistered user can view listings near an institution of their choice.
- 1.1.3. An unregistered user can view the Home page.
- 1.1.4. An unregistered user can view the Terms of Service page.
- 1.1.5. An unregistered user can view the FAQ page.
- 1.1.6. An unregistered user can view the Features pages.
- 1.1.7. An unregistered user can view the About page.

1.2. Registered User

- 1.2.1. A registered user should be able to login to with their username and password.
- 1.2.2. A registered user should be able to logout of their account.
- 1.2.3. A registered user should be able to submit a forgotten password form.
- 1.2.4. A registered user should be able to submit a forgotten email form.
- 1.2.5. A registered user should be able to change their username.
- 1.2.6. A registered user should be able to change their email address.
- 1.2.7. A registered user should be able to change their password.

1.3. Students

- 1.3.1. A student user must have a verified edu email.
- 1.3.2. A student user must have a completed profile.
- 1.3.3. A student user shall be able to view the student dashboard.
- 1.3.4. A student user shall be able to view student user profiles.
- 1.3.5. A student user shall be able to message another user.
- 1.3.6. A student user should be able to add other student users to a group.
- 1.3.7. A student user should be able to add a landlord to a group.
- 1.3.8. A student user shall be able to search for listings.
- 1.3.9. A student user shall be able to edit their personality.
- 1.3.10. A student user should be able to edit their schedule.
- 1.3.11. A student user should be able to edit their hobbies.
- 1.3.12. A student user shall be able to filter listings by price.
- 1.3.13. A student user shall be able to filter listings by amenities.
- 1.3.14. A student user shall be able to filter listings by distance from university.
- 1.3.15. A student user shall be able to filter roommate selections by personality.
- 1.3.16. A student user shall be able to filter roommate selections by major.
- 1.3.17. A student user shall be able to filter roommate selections by hobbies.
- 1.3.18. A student user shall be able to filter roommate selections by schedule.
- 1.3.19. A student user should be able to view the location of a listing on a map.
- 1.3.20. A student user shall be able to favorite a listing.

1.4. Landlords

- 1.4.1. A landlord user shall be able to view the landlord dashboard.
- 1.4.2. A landlord user shall be able to view their own profile.
- 1.4.3. A landlord user shall be able to post listings.
- 1.4.4. A landlord user shall be able to edit their listings.
- 1.4.5. A landlord user shall be able to view student profiles.
- 1.4.6. A landlord user will be able to respond to student user messages.
- 1.4.7. A landlord user shall be able to delete their listings.

1.5. Admin

- 1.5.1. Admin users should be able to approve or deny listings created by landlords.

2. Priority 2 - Desired

2.1. Unregistered User

2.2. Registered User

- 2.2.1. A registered user should be able to delete a chat.

2.3. Students

- 2.3.1. A student user should be able to change their university.
- 2.3.2. A student user shall be able to rate landlords.

2.4. Landlords

- 2.4.1. A landlord user shall be able view historical listing data.
- 2.4.2. A landlord user shall be able to repost their listing once expired.

2.5. Admin

- 2.5.1. Admin users should be able to edit registered user profiles.

3. Priority 3 - Opportunistic

3.1. Unregistered User

3.2. Registered User

3.3. Students

3.3.1. A student user shall be able to report a listing.

3.3.2. A student user shall be able to report another user.

3.4. Landlords

3.4.1. A landlord shall be able to report a listing.

3.4.2. A landlord shall be able to report another user.

3.5. Admin

3.5.1. Admin users should be able to disable non-admin user accounts.

3.5.2. Admin users should be able to remove a listing.

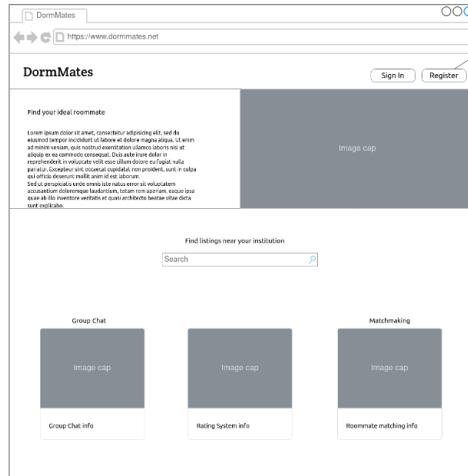
3.5.3. Admin users should be able to view reports.

3.5.4. Admin users should be able to respond to reports.

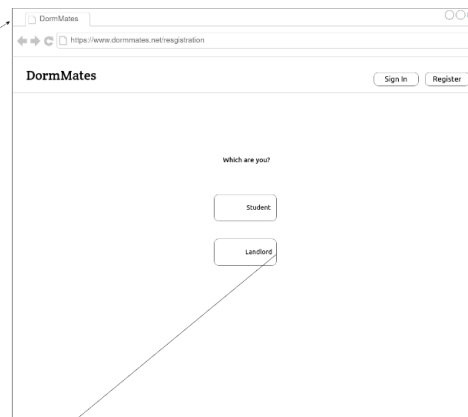
Wireframes

Use Case 1: Creating a Landlord Account and Posting a Listing

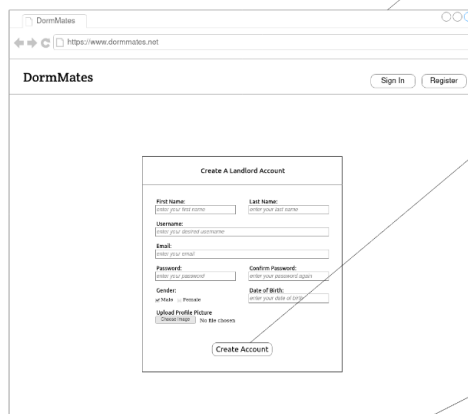
1. General user visits the website for the first time with the intention of creating an account.



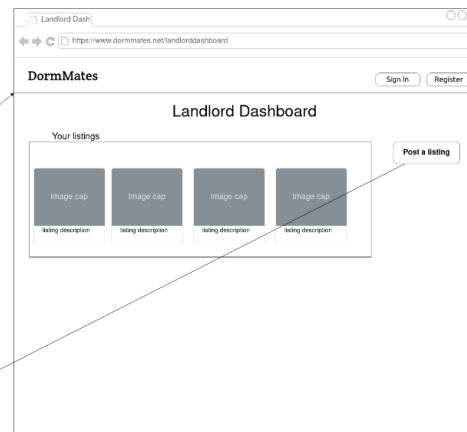
2. User selects the type of account they want to create



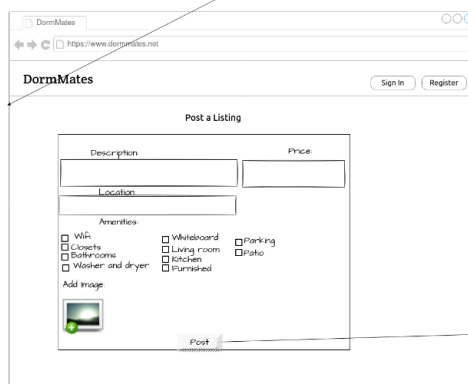
3. User creates a landlord account



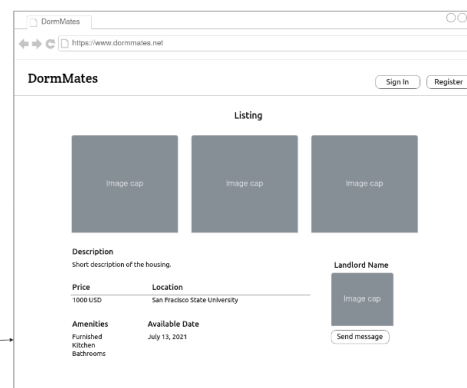
4. Landlord dashboard allows landlord users to post new listings



5. Form that allows landlord users to create a new listing

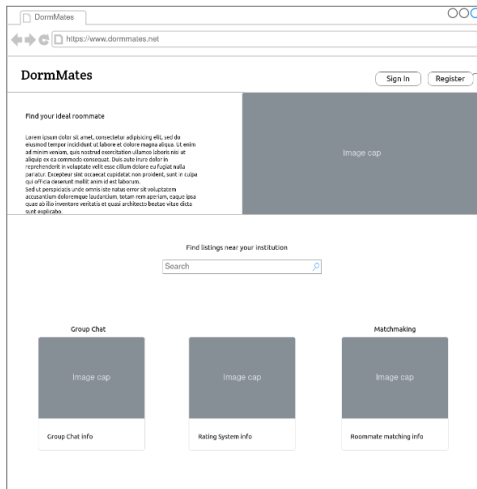


6. After posting a listing, landlord users can view their listing on a dedicated page

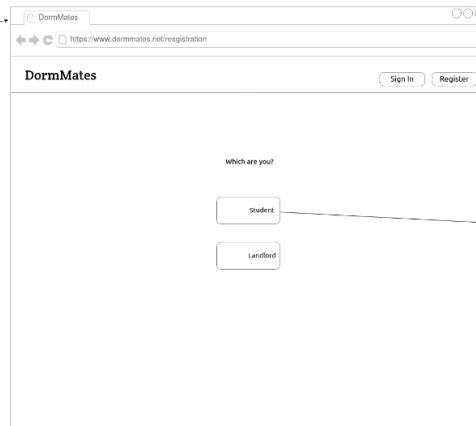


Use Case 4: Student Creates a New Account

1. General user visits the website for the first time with the intention of creating an account.



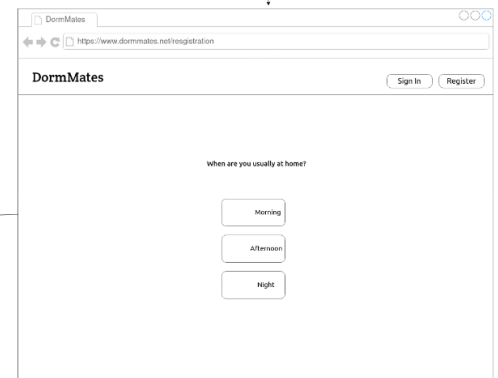
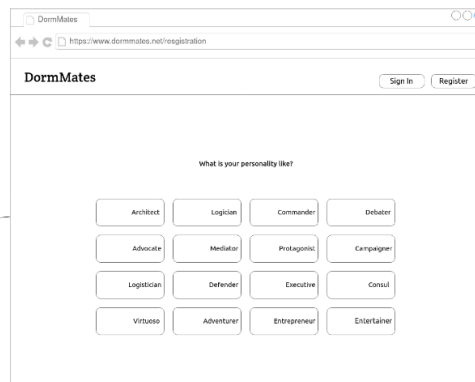
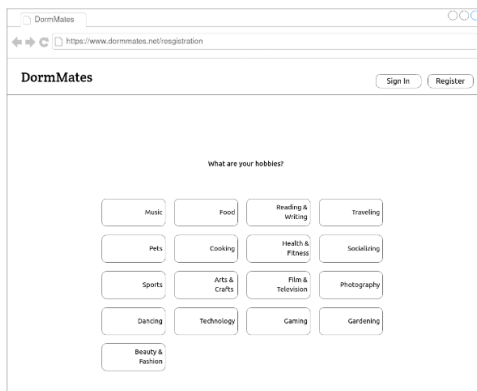
2. User selects the type of account they want to create



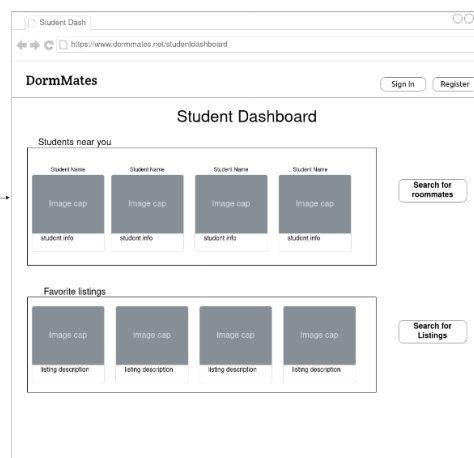
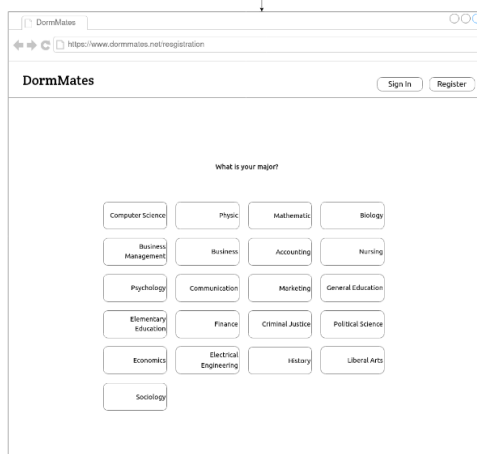
3. User creates a student account



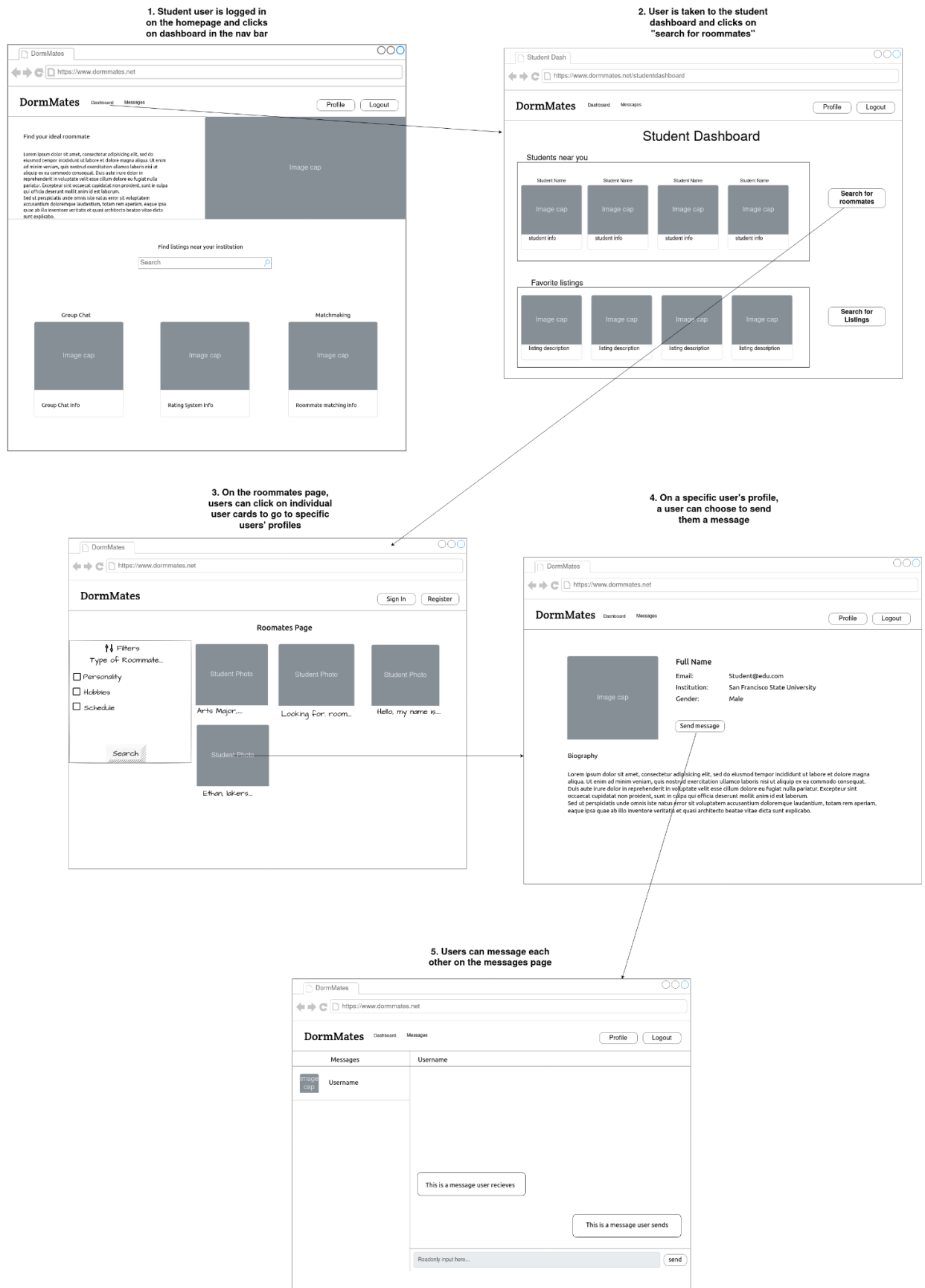
4. After creating a student account, student users must complete a questionnaire



5. When student users complete their questionnaire, they get recommendations for other students based on their questionnaire answers on the student dashboard



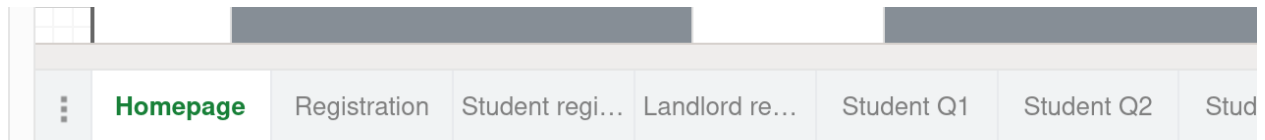
Use Case 5: Student Needs Roommates



All Wireframes

Each individual wireframe can be viewed [here](#).

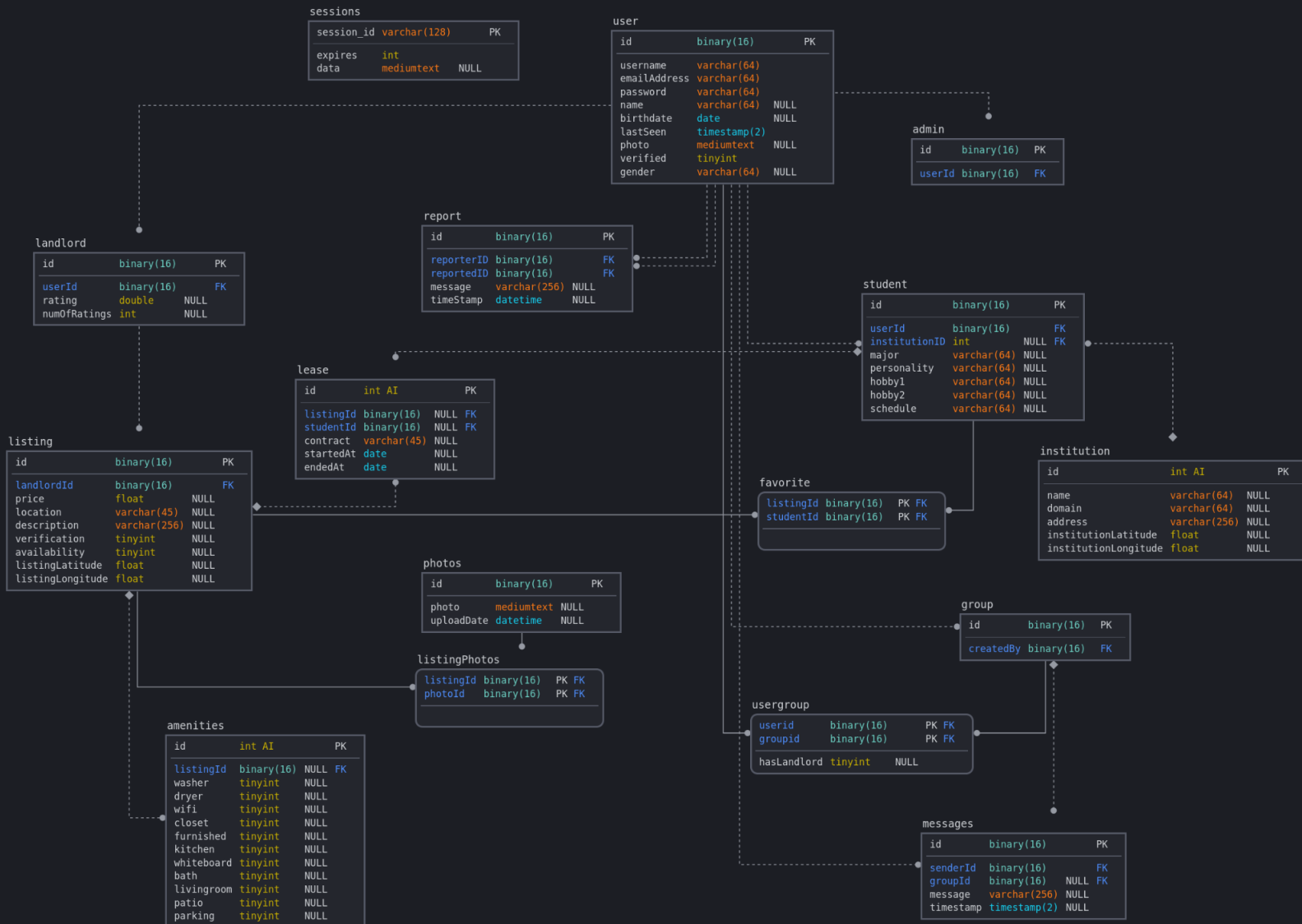
Each tab at the bottom of the web page represents one wireframe.



High Level Database Architecture and Organization V2

Database Model

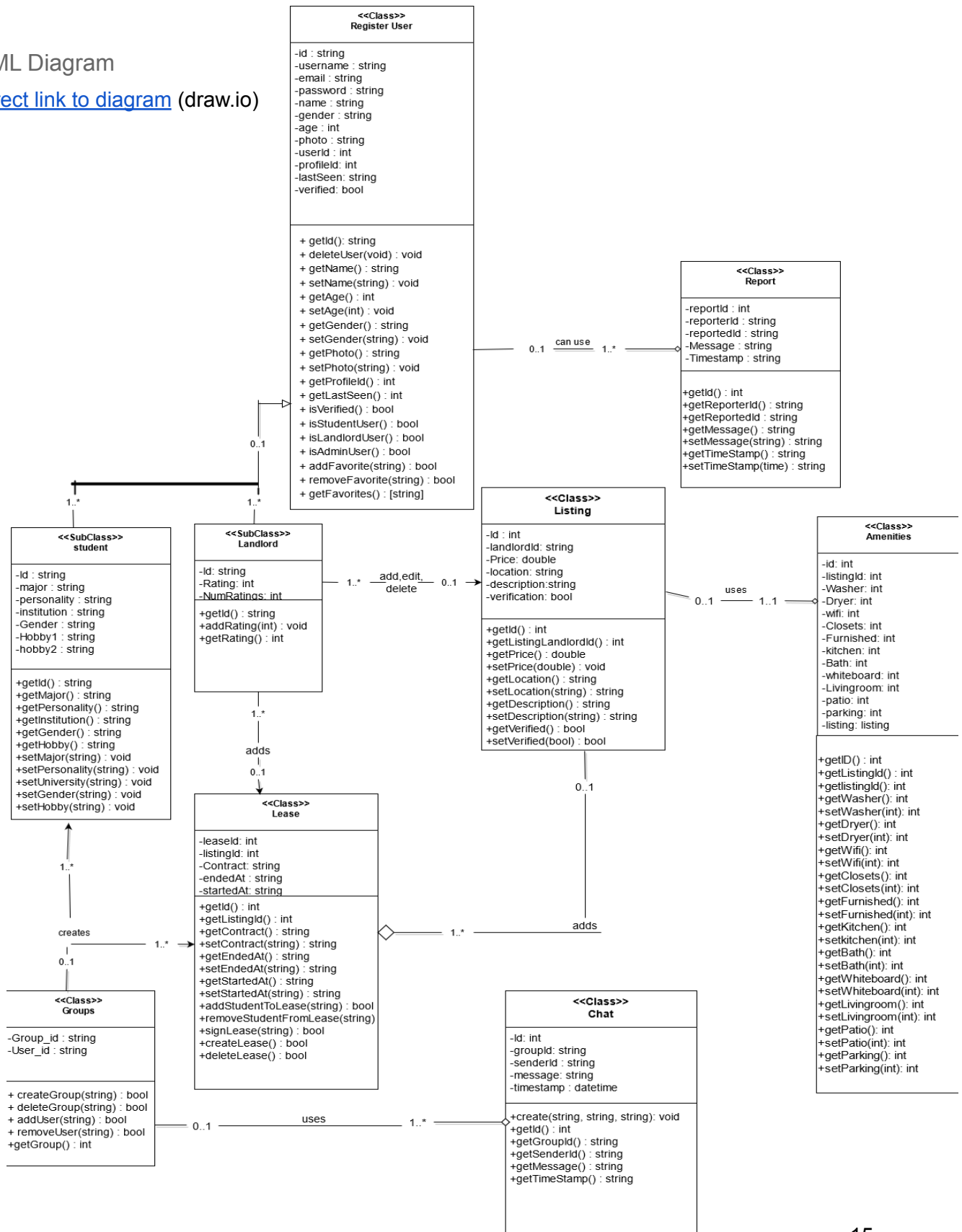
[DB SQL](#) (pastebin.com)



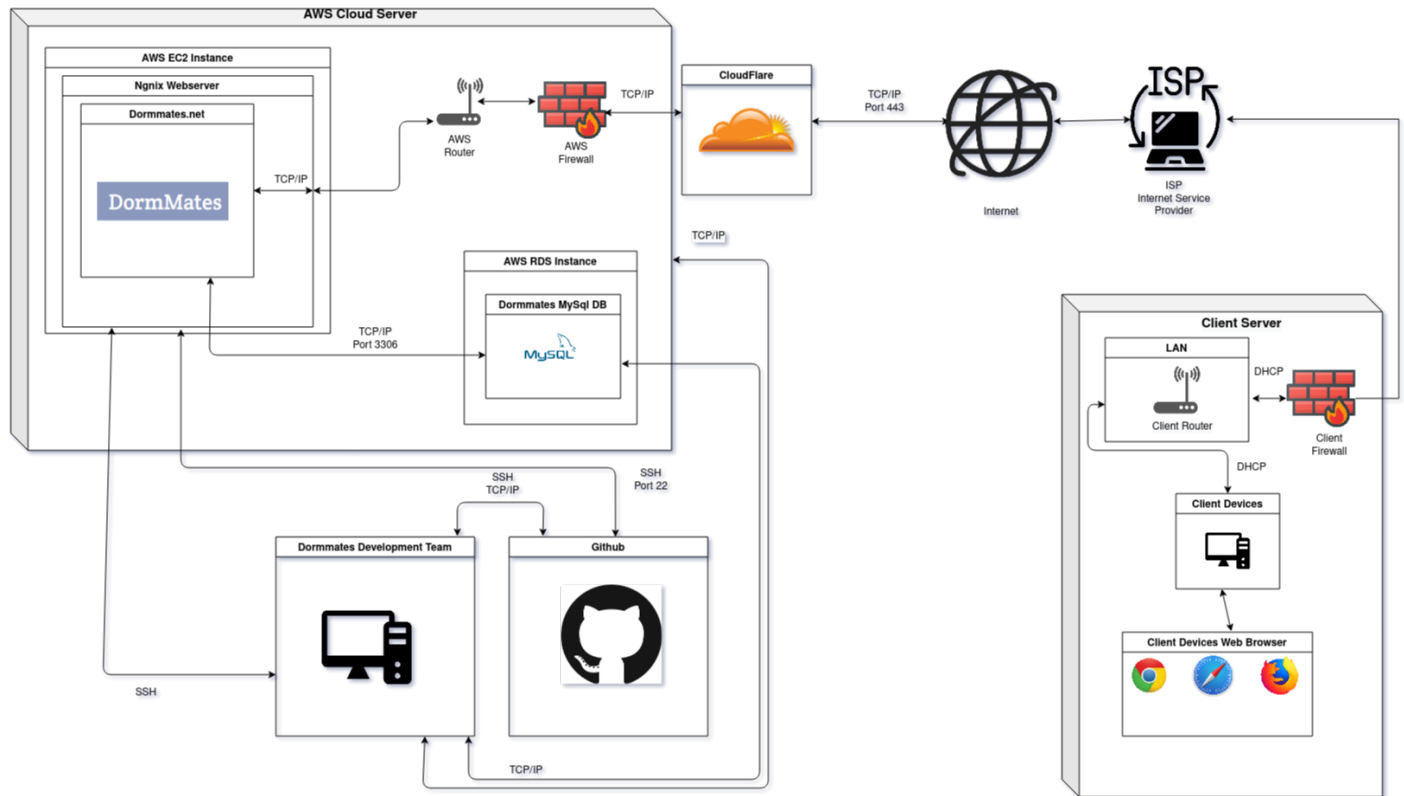
High Level Diagrams V2

UML Diagram

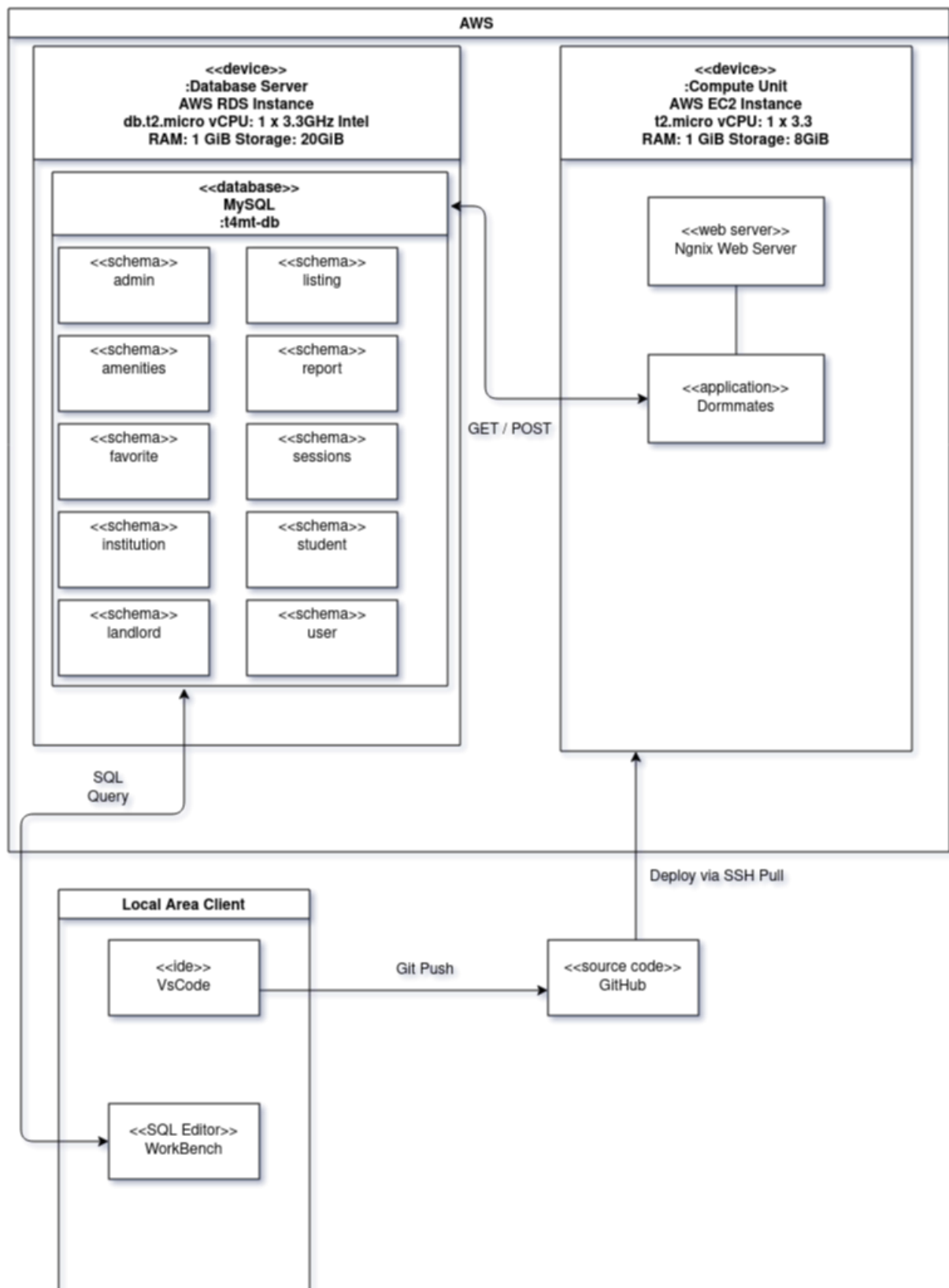
[Direct link to diagram \(draw.io\)](#)



Network Diagram



Deployment Diagram



Detailed List of Contributions

Andrei

- Hosted 1 team meeting where we discussed M2 feedback and worked on functional requirements V3.
- Created Jira tasks to address M2 feedback.
- Created Jira tasks for the frontend team to complete the horizontal prototype.
- Created Jira tasks for the backend team to work on numerous backend APIs.
- Hosted pair programming sessions with the backend team where we discussed implementation of key backend services.
- Hosted pair programming sessions with the frontend team where we discussed implementing wireframes.
- Gave feedback to the frontend and backend teams on their respective tasks.
- Performed code reviews on both the frontend and the backend.

Jonathan

- Participated in team meetings and worked on functional requirements V3.
- Worked on addressing business rules feedback.
- Hosted many pair programming sessions throughout the week with the backend team in order to implement key areas of the backend.
- Created backend documentation for the listing and leases API.
- Implemented the listing API and the lease API.
- Tested the listing API, lease API, search API.
- Assisted in testing and creation of search and favorites API.
- Updated and implemented Database models

Meeka

- Participated in team meetings and worked on functional requirements V3.
- Hosted pair programming sessions with the frontend team.
- Worked on addressing the vertical prototype feedback.
- Created wireframes for the profile, listing, and chat pages.
- Converted profile, listing, and chat wireframes into pug views.
- Created CSS used by the frontend team.
- Implemented scripts to make the frontend dynamic.
- Created dummy data that is used by the profile, listing, and chat pug views.

Jimmy

- Participated in team meetings and worked on functional requirements V3.
- Hosted pair programming sessions with the frontend team.
- Worked on addressing the UML diagram feedback.
- Created wireframes for the student and landlord dashboard pages.
- Converted student and landlord dashboard wireframes into pug views.
- Created dummy data that is used by the student and landlord dashboard pug views.

Alexandre

- Participated in team meetings and worked on functional requirements V3.
- Worked on addressing the Network and Deployment diagrams feedback.
- Hosted many pair programming sessions throughout the week with the backend team in order to implement key areas of the backend.
- Created backend documentation for the search and favorites API.
- Implemented the search and favorites API.
- Tested the search and favorites API.
- Assisted in creation of lease API and listing API.
- Assisted in testing lease API and listing API.

Sayed

- Participated in team meetings and worked on functional requirements V3.
- Worked on addressing the entities and ERD feedback.
- Created wireframes for the roommates, listings, and create-a-listing pages.
- Converted roommates, listings, and create-a-listing pages into pug views.
- Created dummy data for the roommates and listings pug views.

Ahad

- Participated in team meetings and worked on functional requirements V3.
- Created backend documentation for the chat and groups API.